

REMARKS

Reconsideration of the present application is respectfully requested.

Summary of the Office Action

Claims 76 and 78 were rejected under 35 U.S.C. § 112 second paragraph, as being indefinite. Claims 37, 39, 42-46, 48-73 and 76-78 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2002/0169926 of Pinckney et al. ("Pinckney"). Claims 40 and 47 were rejected under 35 U.S.C. § 103(a) as being unpatentable based on Pinckney. Claim 41 was rejected under 35 U.S.C. § 103(a) as being unpatentable based on Pinckney in view of U.S. Patent No. 6,744,763 to Jones et al. (hereinafter, "Jones").

Summary of Interview

Applicants thank the Examiner for the courtesy shown during the telephone interview between the Examiner and Applicants' representative (the undersigned) on 9/18/2007. Applicants agree with the summary of the interview as set forth in the Interview Summary mailed by the PTO on 9/26/2007. No agreement was reached.

Summary of Amendments

In this response, claims 37 and 39-78 have been canceled, and claims 79-96 have been added. No claims have been amended. No new matter has been added.

Therefore, claims 79-96 are now pending.

Discussion of Rejections

The present is thought to be novel and non-obvious at least because it includes a network caching system that includes both a protocol *dependent* caching subsystem *and* a protocol *independent* caching subsystem that is distinct from the protocol dependent caching subsystem (see, e.g., specification at paragraphs [75] and [76]; Fig. 3). This architecture is advantageous in that (among other reasons) separation of the storage/retrieval functionality from the rest of the system allows these functions to be generalized and optimized, without consideration of streaming protocol or encoding format (specification, para. [105]).

The protocol dependent caching subsystem includes a plurality of streaming media encoders for different streaming media protocols (specification at para. [107] and is generally responsible for interacting with an upstream server to obtain streaming media data.

The protocol *independent* caching subsystem is responsible for storing and retrieving streaming media data to and from (respectively) non-volatile storage in the cache. The protocol *independent* caching subsystem manages the flow of streaming media data to and from the operating system of the cache system in a streaming media *protocol independent manner* (specification, para. [105]) *however*, it nonetheless stores the streaming media data in a streaming *protocol specific form* (see, e.g., specification at paragraphs [70], [75] and [76]). The protocol independent caching subsystem also is responsible for *pacing* delivery of streaming media data to a client.

Referring now to the claims, claim 79 recites:

79. (New) A network caching system to cache streaming media, comprising:

- a processor;
- a non-volatile mass storage facility coupled to the processor, to cache streaming media data; and
- a memory storing program code, for execution by the processor, the code implementing:

- an operating system;
 - a **protocol dependent** caching subsystem, which includes a plurality of streaming media encoders to support a plurality of streaming media protocols, wherein the protocol dependent caching subsystem is configured to interact with an upstream server, including to obtain from the upstream server streaming media data encoded according to a particular streaming media protocol, to decode the streaming media data for storage in the mass storage facility, and subsequently to encode the streaming media data into a first streaming media protocol used by a client system after the streaming media data is retrieved from the mass storage facility for delivery to the client system; and

- a **protocol independent** caching subsystem, distinct from the protocol dependent caching subsystem, configured **to manage flow of streaming media data to and from the operating system in a streaming media protocol independent manner**, including **to store the streaming media data** obtained from the upstream server in the mass storage facility in a **streaming protocol specific form** after the streaming media data has been decoded by the protocol dependent caching subsystem, and subsequently to retrieve the streaming media data from the mass storage facility for subsequent encoding by the protocol dependent caching subsystem and delivery to the client system, wherein the protocol independent caching subsystem further is configured to deliver the streaming media data to the client system according to the first streaming media protocol **and to control a pace of delivery of the streaming media data to the client.** (Emphasis added.)

Applicants respectfully submit that Pinckney does not disclose or suggest such an apparatus. In particular, Pinckney does not disclose or suggest a network

caching system that includes both a protocol *dependent* caching subsystem *and* a protocol *independent* caching subsystem. Pinckney does not disclose or suggest a network caching system in which the functionality which interacts with the cache's storage facility has been separated out from the functionality which interacts with the upstream server and where the former has been implemented to operate in a protocol *independent* manner. Insofar as Pinckney discloses, the network communication functionality and the cache read/write functionality are *intertwined* and, therefore, protocol *dependent*.

Second, Pinckney does not disclose or suggest (per claim 79) a protocol *independent* caching subsystem which not only interacts with the cache's storage facility, but also *controls a pace of delivery* of the streaming media data to the client. Note that even if (*arguendo*) Pinckney discloses controlling the pace of delivery of streaming media data, that by itself would not satisfy this claim limitation; it must be a protocol *independent* caching subsystem, which interacts with the cache's storage facility in the manner recited, which controls the pace of delivery.

Third, Pinckney does not disclose or suggest that (per claim 79), even though the protocol independent caching subsystem *operates* in a protocol independent manner, it nonetheless *stores* the streaming media data in the storage facility in a streaming *protocol specific form*. To the contrary, Pinckney discloses that streaming media data is *translated* by a protocol translator 36 and then stored *in a protocol-independent (canonical) form* (para. [0032]). The data is then translated back into a protocol-specific form by a protocol translator 38 before being sent to a client. This is

contrary to the present invention, in which the streaming media data is stored in a form that is *streaming media protocol specific*. A significant advantage of the present invention is that it avoids the need for any protocol translation of the sort disclosed in Pinckney (i.e., into canonical form), which tends to be CPU intensive.

In the present Office Action the Examiner states (p. 12):

Examiner cannot find in paragraph [0032] of Pinckney where streaming media content is being translated. It is unclear where the applicant is reading this. In paragraph [0032], Pinckney is manipulating the data that is transmitted with the actual content, not the streaming media content as assumed by applicant.

The last sentence in the Examiner' s above quoted remark is not understood by Applicants (i.e., what is " the data that is transmitted with the actual content?"). Regardless, Applicants respectfully submit that the translation function mentioned in Pinckney is clearly understood by Applicants and is disclosed at the middle of para. [0031] and in the second sentence of para. [0032] as follows:

The SDA 28 should therefore be able to perform a *protocol translation* and/or cache and store files representing the content requested by the client in a *protocol-independent (canonical) form*.

• • •

The SDA 28 includes a protocol translator 36 which *strip [sic] the network headers* from the received packets and *generates protocol-independent canonical payload data packets*.

This disclosure is clearly *contrary* to the claim language of claim 79, i.e., contrary to " stor[ing] the streaming media data obtained from the upstream

server in the mass storage facility in a streaming *protocol specific form*” (emphasis added). Thus, Applicants respectfully submit that Pinckney clearly *teaches away* from the present invention, as claimed, and for at least that reason, cannot anticipate the present invention or render it obvious.

On this point, the Examiner further states (p. 12):

Pinckney teaches, on pages 2-3, paragraph 0031 that content is stored and subsequently transmitted to a client. The content may be in an incompatible protocol as required by the client and therefore protocol conversion or translation would be deemed necessary. *However, if protocol translation or conversion is not required, for example a situation in which the protocol is deemed compatible with the client, then protocol conversion or translation is not required and it would make no sense to perform unnecessary protocol conversion or translation. It is best understood that the steps of protocol translation as discussed in paragraph 0032 on page 3 of Pinckney would only be performed if protocol translation would be deemed necessary.* Therefore, Pinckney teaches the storage of streaming media data wherein the streaming media data is stored in a form that is specific to a particular streaming media protocol. (Emphasis added.)

First, Applicants respectfully submit that the Examiner has made an incorrect assumption about the system/technique in Pinckney. Specifically, it *does* indeed make sense for the system of Pinckney to store the streaming media data in the canonical (protocol-independent) format *regardless* of whether the protocol of the data is compatible with the requesting client, because one of the stated objectives of Pinckney is to be able to *efficiently* stream the same data to *different clients using various different streaming protocols*. See, for example, Pinckney at para. [0069] starting with the last sentence on page 7, second sentence in para. [0068], and first sentence of para. [0006]. If the system of Pinckney were *not* to translate a given set of data into the

canonical format for storage, that would *reduce the efficiency* with which the system could stream that set of data to other clients that use *other (i.e., incompatible) formats*. Hence, even when the protocol is compatible with a particular client, protocol translation is *not* "unnecessary" in the system of Pinckney, as the Examiner contends. As such, the Examiner's assumption is incorrect.

With the above in mind, consider also that Fig. 3 of Pinckney does *not* show any data path that would suggest either protocol translator 36 or protocol translator 38 can be *bypassed* at any time (e.g., when the protocol is compatible with the requesting client), nor is there any hint found in the text that those elements are used *only sometimes*. Since the scenario mentioned by the Examiner (i.e., where the protocol is compatible with a particular requesting client) may be assumed to be fairly commonplace, one must assume that Pinckney thought of it and, therefore, would have shown such a bypass data path and/or at least mentioned that protocol translation is sometimes unnecessary, *if* in fact that had been the approach that Pinckney intended.

To the contrary, however, Applicants respectfully submit that Pinckney at least *clearly implies*, if not explicitly discloses, that streaming media content is *always* translated into the protocol-*independent* (canonical) form before being cached in the SDA (para. [0032]), and that such an approach makes logical sense in the context of that system for the reason stated above. As explained above, that approach is clearly contrary to that of the present invention as claimed.

For at least the above reasons, therefore, Applicants respectfully submit that claim 79 and all claims which depend on it are patentable over the cited.

Independent claims 86 and 90 include limitation similar to those in claim 79 discussed above and, therefore, are also thought to be patentable over the cited art along with their dependent claims, for similar reasons.

Claim 86

In addition, independent claim 86 recites the additional limitation:

wherein the streaming media data are stored in the mass storage facility as a plurality of discrete data objects, **each of said data objects containing a separate portion of a media stream** in the form of a **plurality** of media packets **formatted for a particular streaming media protocol**.

Pinckney does not disclose or suggest storing the streaming media data *as a plurality of discrete data objects, each of which contains a separate portion of a media stream* as a plurality of media packets. And, as discussed above, Pinckney *teaches away* from the idea of storing the streaming media data in the cache in a form that is *formatted for a particular streaming media protocol*. For these additional reasons, therefore, claim 86 and all claims which depend on it are thought to be patentable over the cited art.

Dependent Claims

In view of the above remarks, a specific discussion of the dependent claims is considered to be unnecessary. Therefore, Applicants' silence regarding any dependent

claim is not to be interpreted as agreement with, or acquiescence to, the rejection of such claim or as waiving any argument regarding that claim.

CONCLUSION

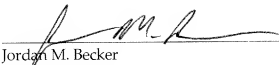
For the foregoing reasons, the present application is believed to be in condition for allowance, and such action is earnestly requested.

If there are any additional charges, please charge Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: November 21, 2007



Jordan M. Becker
Reg. No. 39,602

Customer No. 48102
1279 Oakmead Parkway
Sunnyvale, CA 94085
(408) 720-8300